Data-Driven Design for Anomaly Detection in Network Access Control Systems

1st Musa Abubakar Muhammad
Faculty of Engineering, Environment and Computing
Coventry University
Coventry, UK
0000-0003-4173-533X

2nd Fabio Caraffini
Department of Computer Science
Swansea University
Swansea, UK
0000-0001-9199-7368

3rd Jarrad Morden
School of Computer Science and Informatics
De Montfort University
Leicester, UK
ORCID or jarrad.n.morden@dmu.ac.uk

4th Adebamigbe Fasanmade
School of Computer Science and Informatics
De Montfort University
Leicester, UK
alex.fasanmade@dmu.ac.uk or ORCID

***Abstract*—Current network access control systems can contain unpredictable interactions between multiple device models, multiple network protocol layers (e.g. TCP, UDP and ICMP), hardware, and clock-skew-specific influences, and cannot detect or identify abnormal behaviours based on the type of device. To complicate things further, the 'bring your own device' policy is increasing security threats, vulnerabilities, and risks to enterprise network environments, making intrusion detection and prevention systems unable to detect illegal and unauthorised access to devices in the enterprise network. The consequences can be disastrous. In this light, we propose a simple but effective clustering approach capable of separating normal and abnormal network traffic patterns to detect such challenges (anomalies). We apply this approach to single devices and aggregations of data per device type. Additionally, we propose plotting the notched box for each cluster to acquire a better understanding of their data distributions and measuring the clusters' performance using the Adjusted Rand Index. Our results show that the proposed method is valid, can be used in several contexts, and features a 95% confidence that most single device and device type distributions overlap, which makes them equivalently usable for anomaly detection purposes.**

***Index Terms*—Data-driven research, Behaviour Profiling, Device Fingerprinting, Network Access Control, K-Means Clustering, Anomaly Detection**

I. Introduction

Data-driven research uses access to big data to extract and discover new knowledge and solve specific problems in a subject domain. This approach is widely investigated and used by organisations, as it allows them to exploit data to model and understand the behaviour of complex systems that could be too challenging to study otherwise [1]. Such systems are usually based on several interacting components with deep linkages and interconnections; they may be designed and synthesised to better understand how physical and natural systems work in the real world. For example, the Internet of Things (IoT) connects sensors and actuators to transport data over a network without requiring human-to-human interaction [2]. Other interesting examples from the computing fields are those described in [3], where a statistical model is used to predict the location of taxis (without prior knowledge of driver behaviour) to study the convergence of road networks, and in [4] and [5], where mobile network traffic data are analysed to determine 3G / LTE mobile traffic patterns based on unpredictability, regularity, backhaul flooding, and request satisfaction.

Due to the unpredictable nature of 'Bring Your Own Device' (BYOD) enterprise networks, the application of data-driven approaches is becoming a complex but interesting research field due to the influence that sensors and actuators have on the collected data. In this light, Network Access Control (NAC) systems are the key to the authentication and authorisation of IoT devices [6]. The authentication and authorisation processes are based on the IEEE 802.1x standard, which dictates the authentication policies for devices connected to an enterprise network. IEEE 802.1x improves the NAC authentication procedure across wired or wireless access points by using the Extensible Authentication Protocol (EAP). To authenticate devices accessing the network in 802.1x, the three components of the supplicant, the RADIUS server, and the Authenticator is used, where the supplicant is the user or client seeking network access, the RADIUS server is the server authenticating the client, and the Authenticator is the wireless access point to which the device is attempting to connect. Despite the complexity and NAC systems, they are susceptible to attacks and intrusions. Therefore, being able to detect anomalies in such systems is the key to preventing serious damage from occurring.

In this article, we focus on anomaly detection and employ a data-driven design approach based on the well-established Knowledge Discovery (KD) and data mining process [7]. Note that the literature contains discordant variants for the KD process, which have been a source of contention among researchers. Some describe a 5, 6, or 9-step method, all of which are similar in terms of processes, except for a few variations at the data pre-processing stage. The latter is

considered a one-step process by many, while others describe it as a four-step procedure; for a complete review, we refer to [8] and [9]. For this study, we consider the most relevant aspects of the available KD processes, consisting of 1) understanding the domain problem; 2) understanding the data; 3) data pre-processing, 4) data mining, and 5) evaluation of the discovered knowledge.

The remaining part of this article is organised as follows: 1) Section II highlights previous related works; 2) Section III provides an overview of the procedures involved in data comprehension and identification; 3) Section IV presents the data used and the corresponding pre-processing process for this study; 4) section V describes the use of the k-mean algorithm in the context of this study; 5) Section VI analyses the results; 6) Sections VII and VIII show validation and evaluation of the proposed data mining process; Section IX draws the conjecture of our research.

II. Related Works

Anomaly detection is a technique for locating and identifying data items that do not match the rest of the data in a given data set [10]. It has been used in a variety of data-driven research contexts, including network intrusion detection, bank fraud detection and medical concerns [11], [12].

Numerous research projects in the field of anomaly detection seek to identify patterns that differ from the rest of the data using well-known machine learning and statistical approaches [13]–[15] falling into three main categories, referred to as `statistics-based`, `classification-based` and `clustering-based` methods.

`statistics-based` anomaly detection approaches apply statistical models such as threshold-based methods, see [16]–[19], and principal component analysis (PCA), see [20]–[23] to perform the detection process. When threshold-based methods are used, the features are initially selected, and a suitable statistical model is applied to their distributions to distinguish between normal and abnormal network traffic. Here, the main drawback is that prior knowledge of abnormal behaviours must be available to set the most appropriate threshold. In contrast, systems based on Principle Component Analysis (PCA), as the on in [21], separate network traffic into two types to discover irregular patterns: traffic data associated with the principal components are classified as anomalous, while the remaining data as typical network behaviour. A thorough discussion of the benefits and drawbacks of the two methods for anomaly detection is presented in [22]. It is interesting to note that although PCA-based anomaly detection requires no prior knowledge, and is capable of reporting dramatic changes after a lengthy period of monitoring, its performance is highly dependent on the degree of coherence of the traffic data.

There are various `classification-based` approaches, including rule-based classification, decision trees, Bayesian networks, neural networks, and support vector machines (SVM) [23], to name a few. Regardless of the machine learning techniques employed, they all consist of training a classifier, mainly on labelled training data, to learn data patterns to then validate the method before using it to detect abnormal traffic. These `classification-based` anomaly detection systems are popular due to their substantial flexibility and high detection probability; nevertheless, they require labelled training data, which must be obtained from network traffic and are often difficult to obtain.

In this light, clustering algorithms are extensively employed in the identification of anomalies, neither requiring pre-labelled data nor needing prior knowledge. A worthwhile system is the one presented in [24], which uses hierarchical clustering to categorise mobile call profiles to discover abnormal traffic behaviours. In this study, the authors compared hierarchical clustering with k-means clustering and discovered a lower temporal complexity in the latter algorithm. A limitation of clustering algorithms is that they overlook regional differences in traffic patterns, which sometimes makes some anomalies difficult to identify.

Hybrid approaches represent the most modern investigation lines in anomaly detection. For example, in [25], a hybrid model using a Deep Auto-Encoder (DAE) in combination with the ensemble K nearest neighbours (K-NN) is used to identify outliers in high-dimensional data. In fact, this mechanism is capable of reducing the dimensionality of the data before applying the K-NN clustering algorithm. A different approach is presented in [26], where different weights are assigned to the data samples to mitigate the negative effect of the unbalanced log sample distribution on the accuracy of the K-NN algorithm. Another research in [27] proposes a combination of the K-NN and Local Outlier Factor (LOF) algorithms to detect anomalous behaviour. The authors in [28] create an anomaly detection model for identifying widespread DNS abnormalities in unsupervised learning using multi-enterprise network traffic data that do not include attack labels (NetFlow data set). This approach calculates the model's detection rate using two clustering methods. The k-means clustering and Gaussian Mixture Model (GMM) approaches are investigated for their great sensitivity in detecting irregularities. In contrast to previous studies, [15] provides an innovative strategy to identify area-specific traffic in a city. By aggregating area units with comparable traffic patterns, anomalous activities within the grouped regions are detected.

In our work, we also propose a new technique where data are divided by using k-means clustering to make it possible to separate normal and anomalous network traffic patterns. In addition, our approach plots the notched box for each cluster to acquire a better understanding of their data distributions and measures the clustering anomaly detection performance using the Adjusted Rand Index.

III. Understanding the data

In this session, we consider and evaluate established repositories, including the Community Resource for Archiving Wireless Data at Dartmouth (CRAWDAD), the Centre for Applied Internet Data Analysis (CAIDA), Outlier Detection data sets (ODDS), DARPA, and MAWI.

We focus on attributes that allow us to characterise and classify the behavioural patterns of the data. Hence, despite being widely used within the research community, we observe that the DARPA data set [29] would not add sufficient value to our approach, as the devices used to generate it are no longer up to date and therefore inadequate to create robust behavioural models capable of dealing with modern attacks. The ODDS data set repository [30] instead offers a larger collection that can be used to address challenges related to anomaly detection. Although this repository has a sufficient number of data sets, they are too deemed unsuitable for this investigation, since they were randomly created from network devices, making it difficult to distinguish the data from smartphones, tablets, and laptops. In CAIDA [31], numerous packets were lost, which can significantly affect the implementation of this investigation. Similarly, the MIT reality data set [32], is also not up to date as the devices used were 2004 models with Symbian OS, which is no longer in use, and its main focus is mobile device usage and movements.

In this light, we use the Crawdad repository [33], where packet interarrival time measurements of 27 different mobile devices (e.g., smartphones, tablets, and laptops) are available.

A. Gatech data set Description

The Gatech fingerprint data set is publicly available in [33] and is provided by the Georgia Institute of Technology as part of their device and device type fingerprint identification research. The Gatech fingerprint data set contains packet Inter-Arrival Time (IAT) values for 27 mobile devices measured on *active*, *passive*, and *isolated* network monitors. The IAT enables a statistical analysis of the reproducible pattern of devices (signals) that measures the delays between successive packets and thus characterises the rate of traffic flow. There are three types of test bed measurements: passive, active, and isolated. The former was formed in a real-world network context, whereas the others were generated in a completely isolated setting with no external influences. The data is organised in different files in the `.mat` format, which is only suitable for use in MATLAB. From these files, we select and extract data from smartphones, tablets, and laptops and convert them into the comma-separated value (CSV) format. The reason for considering only these devices is that this piece of research targets NAC systems. Thus, we are limited to selecting only supported devices. Conversion to CSV is convenient, as most data mining tools, such as `RapidMiner Studio` [34], do not support the `.mat` file format. Finally, we use the following format to organise our data: `filename`→`Application`→`Protocol`→`Case` (e.g. iPerf-TCP-Case2) to make it easy to recognise during our experiments.

B. Overview of the prepared data sets

The *active* data set contains the packet IAT data of 68 devices, namely 10 Acer netbooks, 10 Asus netbooks, 8 Gateway netbooks, 2 Google phones, 2 Lenovo laptops and 2 tablets. Network traffic is measured in an active network environment, allowing devices to flow at their natural rate without passing through an encrypted channel. This network traffic is generated randomly from the mentioned devices with a distribution rate in the range $0 - 200$ kbps repeatedly for 5 seconds. The isolated data set comprises packet IAT data from 94 mobile devices, including five Dell netbooks, three iPads, two iPhone 4G models, two iPhone 3G models, and two Nokia phones. Network traffic was measured in a completely isolated network environment without radio frequency (RF) leakage or interference.

The *isolated* data set comprises packet IAT data from 94 mobile devices, including 5 Dell netbooks, 3 iPads, 2 iPhone 4G models, 2 iPhone 3G models, and 2 Nokia phones. The network traffic was measured in a completely isolated network environment free of radio frequency (RF) leakage or interference.

The *passive* data set contains the IAT data of 245 devices, including 10 Acer netbooks, 10 Asus netbooks, 8 Gateway netbooks, 2 Google phones, 2 Lenovo laptops, and 2 tablets. The network traffic measure is similar to that of the *active* network traffic, except that it is measured using passive network monitors.

IV. Data Pre-processing

We preprocessed the *active*, *isolated*, and *passive* data sets in `RapidMiner Studio` [35] with the three-block system graphically represented in Figure 1. Each block follows the `RapidMiner Studio` implementation of the 1) `load data` operator; `k-means` clustering operator; and 2) the cluster model `visualiser` operator.

Fig. 1: From left to right, the `load data`, `k-means` and `visualiser` operators forming thr data preprocessing system built in `RapidMiner Studio`.

Each connected operator has its own configuration settings. The `load data` operator accesses the data sets stored in the repository and loads them into the process. The `k-means` operator is set with the default parameters 10 and 100 for the maximum number of runs and the maximum number of optimisation steps and $K = 2$, respectively. The measure type used for the clustering process is the mixed nominal, numerical, and Bregman divergence, which works hand in hand with the divergence option so that when the measure type changes, the options in the divergence change. For example, when a nominal value is selected, the divergence is set to the nominal distance; when a Bregman divergence is selected, many options can be used to calculate the cluster distance. The well-known examples of the cluster distance measures are the squared Euclidean distance, the Mahalanobis distance, and the squared loss, among others, and these are available under the divergence tab (all implemented in `RapidMiner Studio`).

Finally, the `visualiser` operator is added to display the clustering results and calculate the essential metrics of each cluster, such as the Davies-Bouldin performance index. With a trial-and-error approach, we unwind the best configuration of parameters.

V. Data Mining using K-means clustering

`k-means` clustering is a frequently used technique for spotting outliers. There are several advantages to this system, including its speed, robustness, and relative efficiency [36]. It is very simple to use and may be combined with iterative refining to provide better results when the data set is distinct or the data are well segregated from one another [37]. When the resultant group in the data is unknown, `k-means` is used to locate a group of similar patterns that exist in the data to generate cluster centroids. These are allocated to each data point to create a new training data set for machine learning classification [38].

A. The k-means Clustering Algorithm

The name of the `k-means` algorithm comes from its mode of operation. The method divides the observations into groups of K (K being an input parameter). It then assigns each observation to a cluster based on the observations that are closest to the cluster's mean (any metric can be used, but the Euclidean distance is the most common). The cluster mean is then recalculated, and the procedure is repeated until convergence is reached. The algorithm operates as follows:

1) Randomly select K points (the means) to centre the clusters;
2) assign each data point to the cluster whose centre is closest (in terms of Euclidean distance by default) to the point;
3) recalculate the centres as the average of the cluster's points;
4) repeat steps 2 and 3 until all clusters have converged (i.e., either clusters stop changing or the centres do not move significantly[1]).

Determining the optimal number of clusters K is a key but challenging factor [36]. The correct choice is usually not clear and depends on the scale and shape of the data points and the desired clustering analysis. Increasing k can reduce clustering errors to the extent that no error can be identified when each data point becomes a single cluster (i.e., when k equals the number of data points).

B. On K-means Parameter Settings

Numerous ways to determine K have been proposed [39], such as the *elbow* method, the *Davies-Bouldin* (DB) index, the *Silhouette* index, the *Dunn* index and the partition coefficient, among others. The DB index is used in several studies [40]–[42], and consists of calculating intracluster similarities and intercluster differences to produce a set of indexed clusters [43]. Setting k equal to the smallest index value is often an effective approach [44].

For these reasons, in this study, we determine the values of the DB indexes of the configurations described in Section IV, and report them in Tables I, II and III. Clearly, in our case, k = 2 seems to be the best option.

TABLE I: Davies-Bouldin index for sample devices in the active network traffic data set

Device	K=2	K=3	K=4	K=5
Acer 1	0.037	0.071	0.376	0.409
Acer 2	0.417	0.153	0.376	0.300
Acer 3	0.067	0.071	0.261	0.385
Acer 4	0.112	0.134	0.175	0.309
Acer 5	0.001	0.027	0.025	0.234
Acer 6	0.071	0.173	0.311	0.412
Acer 7	0.408	0.225	0.339	0.206
Acer 8	0.068	0.070	0.199	0.378
Acer 9	0.408	0.154	0.202	0.281
Acer 10	0.035	0.046	0.276	0.248
Acer	**0.001**	**0.060**	**0.075**	**0.188**

TABLE II: Davies-Bouldin index for sample devices in the isolated network traffic data set

Device	K=2	K=3	K=4	K=5
Dell 1	0.001	0.346	0.234	0.280
Dell 2	0.265	0.366	0.355	0.507
Dell 3	0.181	0.349	0.337	0.256
Dell 5	0.291	0.330	0.349	0.386
Dell	**0.349**	**0.386**	**0.330**	**0.291**
iPad 1	0.046	0.346	0.234	0.280
iPad 2	0.265	0.366	0.355	0.507
iPad 3	0.181	0.349	0.256	0.337
iPads	**0.307**	**0.416**	**0.393**	**0.367**
iPhone 3G 1	0.326	0.375	0.358	0.393
iPhone 3G 2	0.074	0.276	0.350	0.365
iPhone 3G	**0.082**	**0.312**	**0.398**	**0.415**
Nokia 1	0.194	0.354	0.180	0.313
Nokia 2	0.174	0.259	0.407	0.263
Nokia	**0.169**	**0.228**	**0.236**	**0.333**

Note that we used a maximum number of runs = 10, the *Bregman* divergence condition with the squared Euclidean distance metric, and a maximum number of optimisation steps = 100.

The studies are performed on sample data sets from the *active*, *isolated* and *passive* network traffic data sets. These were measured using different network protocols, such as the Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP). This helps identify the presence of differences from these protocols.

[1]A threshold must be pre-defined, but note that a maximum number of allowed iteration must be set as well or practical reasons.

TABLE III: Davies-Bouldin index for sample devices in the passive network traffic data set

Device	K=2	K=3	K=4	K=5
Gateway 1	0.030	0.118	0.285	0.325
Gateway 2	0.077	0.383	0.442	0.375
Gateway 3	0.034	0.347	0.325	0.361
Gateway 4	0.135	0.351	0.430	0.370
Gateway 5	0.037	0.317	0.303	0.392
Gateway 6	0.030	0.156	0.321	0.379
Gateway 7	0.034	0.204	0.334	0.001
Gateway 8	0.032	0.385	0.340	0.390
Gateway	**0.047**	**0.329**	**0.312**	**0.365**

VI. Analysis of the clustering Results

This section reports the `k-means` clustering results for the TCP, UDP and ICMP protocols, and their device types, for each network traffic sample under study. The analysis is based on their corresponding centroid points after applying the clustering algorithm to the devices individually. Subsequently, these results are also aggregated per device type. For example, data from Acer devices 1 to 10 are concatenated into one data file to see if different patterns emerge from the data sets; this applies to all devices used throughout. Note that this centroid point analysis helps in the creation of clusters that can be used to identify and analyse the relationships between the inter-arrival time (IAT) values for each sample device in the data sets.

A. Active Network Traffic data set

The `k-means` results for Acer netbooks 1 to 10 in the *active* network traffic data set *active* presented in Table IV show that the centroid points in the first cluster C_0 fall at $0.009s$, which has IAT data points associated with it. It can be detected that, for example, there are $395,457$ IAT points associated with C_0 for the Acer 1 device. for the other Acer $2-10$, we observe inter-arrival points in the range $395,847-398,019$. These results show that all devices have a similar pattern in C_0 and the same analysis can be applied to the other devices, such as Asus, Gateway and Lenovo, although they present similar IAT values in C_0.

Regarding the second cluster C_1,we observe a small number of IAT points, where the minimum and maximum values are 29 and 56, respectively. Furthermore, all devices under consideration have different IAT values between $0.694-0.990s$, except Acer 5, which has $5.744s$. The C_1 results for Acer netbooks $1-4$ and $6-10$ show that they have smaller IAT points, with the smallest being 29 for Acer 10, and the largest being 54 and 56 for Acer 2 and 9 respectively. The device type **Acer** has $3,968,591$ and 1 IAT points in C_0 and C_1, respectively. The rest of the devices have 31 and 34 IAT points.

This analysis for C_0 and C_1 shows that there are clear mean differences between the clusters for each device, which can be used for outlier detection algorithms to detect abnormal patterns located within the data points in each cluster. For example, Acer 5 has an inter-arrival time point value of $5.744s$ in one of the data points, but there are still points in the data that were not outliers because this largest value influenced C_1 for the second partition of the device type **Acer**, and the algorithm is left with the option to partition that point.

TABLE IV: Descriptive analysis of sample device types for the Ping ICMP active network traffic data set

Device	Cluster	IAT Points	Centroid
Acer 1	C_0	395,457	0.009
	C_1	33	0.990
Acer 2	C_0	396,464	0.009
	C_1	54	0.706
Acer 3	C_0	395,847	0.009
	C_1	64	0.970
Acer 4	C_0	397,180	0.009
	C_1	31	0.943
Acer 5	C_0	396,442	0.009
	C_1	1	5.744
Acer 6	C_0	396,316	0.009
	C_1	33	0.968
Acer 7	C_0	397,669	0.009
	C_1	31	0.694
Acer 8	C_0	397,266	0.009
	C_1	34	0.971
Acer 9	C_0	398,019	0.009
	C_1	56	0.696
Acer 10	C_0	397,566	0.009
	C_1	29	0.990
Acer	C_0	**3,968,591**	**0.009**
	C_1	**1**	**5.744**

B. Isolated Network Traffic data set

Similarly, we report the results for Dell devices 1 to 5, iPad 1 to 3, and Nokia phones 1 to 2 in the *isolated* network traffic data sets presented in Table V. Here, the centroid points for all devices in the first cluster C_0 fall at $0.001s$. Each device has inter-arrival points associated with C_0 and C_1. Note that the IAT points associated with C_0 for Dell 1 are $840,955$ and for C_1 are 344. The remaining Dell $2-5$ devices display IAT points between $1,327,228$ and $3,059,230$. For iPads $1-3$, these are between $840,955$ and $1,327,118$, while the Nokia $1-2$ devices have IAT points distributed between $844,462$ and $1,562,924$.

As previously done, the device types in boldface, i.e. **Dell**, **iPads** and **Nokia** are the concatenation of the total inter-arrival time points for Dell $1-5$, iPad $1-3$ and Nokia $1-2$. Note that IAT in C_0 and C_1 for **Dell** is observed in $9,099,736$ and 588, showing that there are some variations in their

IAT distributions. This means that with further investigation, abnormalities must be found for the investigation.

In C_1, devices have different IAT values. Dell 4 has the smallest centroid value (i.e. $0.004s$), while Dell 5 has the maximum centroid value of $0.428s$, while Dell $1-3$ was observed with 0.074 to $0.132s$. As for iPads, the centroid point values lie between 0.074 and $0.132s$, in which iPad 2 and 1 have the minimum and maximum IAT values, respectively. Moreover, the Nokia phone has the highest centroid values compared to all other devices in the data sets - with Nokia 1 was having $1.593s$ and Nokia 2 $2.863s$. This shows that there is a high tendency of anomalies in their inter-arrival time distributions. The C_1 centroid point values for the device type **Dell**, **iPads** and **Nokia** have IATs of $0.428s$, $0.111s$ and $2.863s$, respectively. Furthermore, a similar analysis can be applied to the remaining devices and their device types in the *isolated* network traffic data sets.

TABLE V: Descriptive analysis of sample devices for the iPerf TCP isolated network traffic data set

Device	Cluster	IAT Points	Centroid
Dell 1	C_0	840,955	0.001
	C_1	344	0.132
Dell 2	C_0	1,327,118	0.001
	C_1	320	0.074
Dell 3	C_0	1,288,629	0.001
	C_1	66	0.111
Dell 4	C_0	2,557,115	0.001
	C_1	26,530	0.007
Dell 5	C_0	3,059,230	0.001
	C_1	17	0.428
Dell	C_0	**9,099,736**	**0.001**
	C_1	**588**	**0.428**
iPad 1	C_0	840,955	0.001
	C_1	344	0.132
iPad 2	C_0	1,327,118	0.001
	C_1	320	0.074
iPad 3	C_0	1,288,629	0.001
	C_1	66	0.111
iPads	C_0	**4,575,663**	**0.001**
	C_1	**5,876**	**0.111**
Nokia 1	C_0	844,462	0.001
	C_1	69	1.593
Nokia 2	C_0	718,428	0.001
	C_1	52	2.863
Nokia	C_0	**1,562,927**	**0.001**
	C_1	**84**	**2.863**

Based on the above results, it becomes clear that there are mean differences between C_0 and C_1 for each device and device type. Although no outliers emerged, this shows that there are significant patterns that can be used to detect outliers from the devices. If these devices are connected to the same enterprise network, intrusion detection will automatically flag Nokia 1 and 2 as abnormal devices because they have the highest centroid values. We aim to build a profile based on the mean differences identified from each of these devices separately so that intrusions can be detected based on a device or device type, depending on the research problem.

C. Passive Network Traffic data set

The results relative to Gateway netbooks $1-8$ in the *isolated* network traffic data sets are displayed in Table VI to show that the centroid points in the first cluster C_0 for all devices of this type fall at $0.011s$. In this case, devices appear to have almost balanced IAT points in the range $320,874 - 320,925$.

Taking into account all the **Gateway** types, IATs are $2,567,066$ and $2,688$ for C_0 and C_1, respectively. In the second cluster C_1, the minimum IAT point of 309 is registered on the Gateway device 8, while the maximum value, i.e. 367, is displayed by Gateway 4. The centroid point values for all devices and their device types lie between 0.128 and $0.135s$.

TABLE VI: Descriptive analysis of sample devices for the iPerf UDP passive network traffic data set

Device	Cluster	IAT Points	Centroid
Gateway 1	C_0	320,880	0.011
	C_1	355	0.135
Gateway 2	C_0	320,913	0.011
	C_1	334	0.131
Gateway 3	C_0	320,929	0.011
	C_1	312	0.134
Gateway 4	C_0	320,803	0.011
	C_1	367	0.128
Gateway 5	C_0	320,886	0.011
	C_1	311	0.135
Gateway 6	C_0	320,925	0.011
	C_1	353	0.135
Gateway 7	C_0	320,848	0.011
	C_1	355	0.135
Gateway 8	C_0	320,874	0.011
	C_1	309	0.135
Gateway	C_0	**2,567,066**	**0.011**
	C_1	**2,688**	**0.135**

These results show that devices have similar patterns in the way they transmit packets within a network, and there are few deviations in their patterns in both C_0 and C_1, since the centroid point values lie between $0.128 - 0.135s$. Also, the same analysis can be applied to other devices in the *passive* network traffic data set.

It should be noted that the above mean differences between clusters can help to gain an in-depth understanding of the IAT values associated with each cluster. Furthermore, they can help detect outliers for devices measured in the *passive* network traffic data set.

VII. Validation of K-means clustering Approach

The above data mining approach is validated using descriptive analysis, a technique used to describe data patterns. To observe the data patterns of each cluster centroid points, we use descriptive statistics on each cluster by generating their corresponding notched box plots and show how the data values for each device and device type are distributed.

A. Validation of the Active Network Traffic data set

Figures 2 and 3 present the notched box plots for C_0 and C_1 for the Acer netbooks described in Section VI-A. Figures show that the notched boxes for both the devices and their device type overlap. This gives a 95% confidence that the devices Acer $1-10$ have the same IAT distribution of the **Acer** device type. Hence, single device or device type can be used further in another data mining approach (e.g. outlier detection) based on devices or device types as appropriate and depending on the problem or research focus.

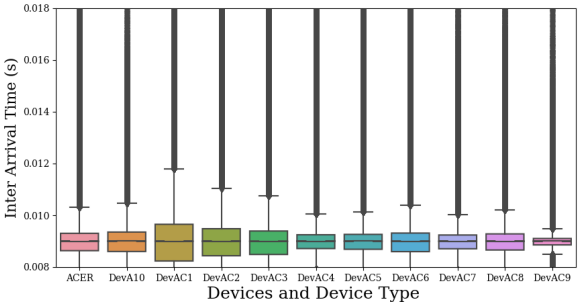

Fig. 2: Notched box plots of C_0 centroid points for the Acer netbook devices $1-10$ and their **Acer** device type on the *active* network traffic data sets

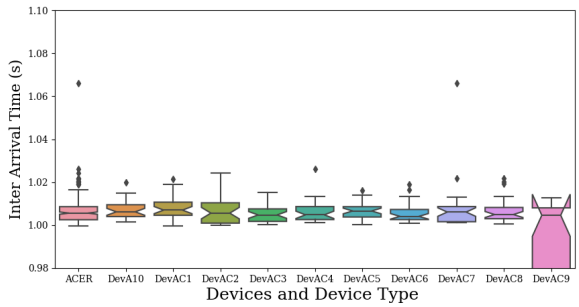

Fig. 3: Notched box plots of C_1 centroid points for the Acer netbook devices $1-10$ and their **Acer** device type on the *active* network traffic data sets

B. Validation of the Isolated Network Traffic data set

The notched plots for Dell netbooks and iPads are similar to those of Acer netbooks and lead to the same conclusions. On the contrary, the boxed boxes for the Nokia phones illustrated in Figures 4 and 5 show that the distribution of single devices and their device type overlap in C_0 and otherwise in C_1. These results mean that the device-type approach cannot be applied to this data set as the IAT values relate to different distributions.

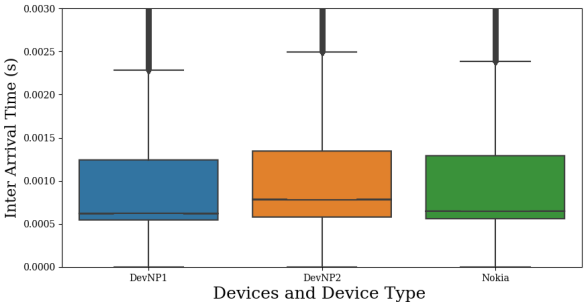

Fig. 4: Notched box plots of C_0 centroid points for Nokia phones and their **Nokia** device type on the *isolated* network traffic data sets

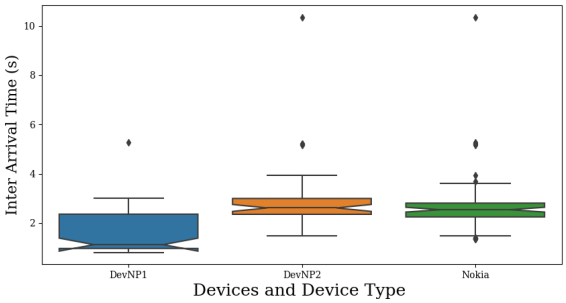

Fig. 5: Notched box plots of C_1 centroid points for Nokia phones and their **Nokia** device type on the *isolated* network traffic data sets

C. Validation of the Passive Network Traffic data set

Also, for Gateway netbooks, the notched boxes presented in Figures 6 and 7 show an overlap of devices and device type distributions, showing 95% confidence that Gateway devices $1-10$ belong to the same IAT distributions of their **Gateway** device type.

Here, we note outliers in both C_0 and C_1 that need further investigation. We envisage applying an outlier detection approach to this data set to understand whether these outlying points are significantly different from the rest of the data values associated with each device and device type in our next study.

VIII. Evaluation of the Discovered Knowledge

To perform KD, we evaluate the performance of the `k-means` clustering algorithm for all device types listed in Tables IV, V and VI to select the best measure of similarities between the clusters and the best performance index of the clusters.

To achieve this, we run the `k-means` with Euclidean distance over the entire data sets from each device type to

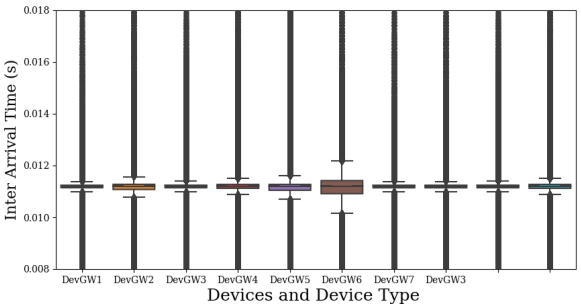

Fig. 6: Notched box plots of C_0 centroid points for Gateway netbooks $1-8$ and their **Gateway** device type on the *passive* network traffic data set

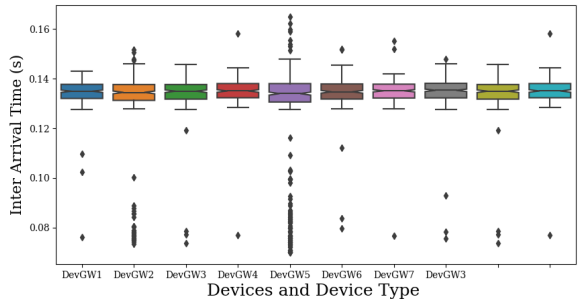

Fig. 7: Notched box plots of C_1 centroid points for Gateway netbooks $1-8$ and their **Gateway** device type in the *passive* network traffic data set

generate 'pseudo-labels'. The stratified cross-validation k-fold (which maintains the imbalance in the data ratio) [38] is applied to pseudo-labels to be able to compute the mean Adjusted Random Index (ARI) score of the IAT points of each k^{th} cluster and the corresponding standard deviation. As folds are added, a portion of the data is used for training, and the rest for testing. Then we fit 80% of the whole data, predict 20%, and report the inertia of the fitted model of the expected randomised index value for each fold.

The ARI ranges from 0 to 1, with 0 indicating that the data clustering is not agreeing on any pair of points and 1 showing that the data clustering is precisely the same.

The results of this procedure, for the device types in the *active* network traffic data set, are shown in table VII. Clearly, the ARI for Acer, Asus and Gateway netbooks indicates that there is a pair agreement between the clusters, with the standard deviation being 0.00. Also, there is a pair agreement between the clusters for Google phones, Lenovo laptops, and tablets, with an ARI of 0.98 and 0.96, which is very close to 1.00, and standard deviations of 0.01 and 0.002, respectively.

Similar results are observed for device types in the *isolated* and *passive* network traffic data set, as presented in Tables VIII and IX, except for Dell netbook and tablet. In fact, the latter have the highest standard deviations, of 0.03 and 0.04, respectively, although the deviation does not affect the performance of these device types.

TABLE VII: The ARI score for device types in the active network traffic data set

Device Type	Mean ARI	St. Dev.
Acer	1.00	0.00
Asus	1.00	0.00
Gateway	1.00	0.00
Google Phone	0.98	0.01
Lenovo	0.96	0.02
Tablet	0.96	0.02

TABLE VIII: The ARI score for the device types in the isolated network traffic data set

Device Type	Mean ARI	St. Dev.
Dell Netbook	0.98	0.03
iPad	1.00	0.00
iPhone 3G	0.99	0.01
iPhone 4G	1.00	0.00
Nokia Phone	0.98	0.02

TABLE IX: The ARI score for the device types in the passive network traffic data set

Device Type	Mean ARI	St. Dev.
Acer	1.00	0.00
Asus	1.00	0.00
Gateway	1.00	0.00
Google Phone	0.93	0.02
Lenovo	1.00	0.00
Tablet	0.90	0.04

IX. Conclusion

We demonstrate a data analysis technique for anomaly detection that uses `k-means` clustering on existing data sets. Instead of using the standard `k-means` technique (scatter-plot), we employ a notched box plot to provide more insight into the data values. We first analyse individual device data to then concatenate all the data from the same device type into one data file and compare the results. This allowed us to understand the relationships between the individual devices and their device types and to demonstrate that their data distributions are similar. Our proposed device type profiling method is a valid approach, featuring a 95% confidence that all the distributions of devices and device types overlap. Our cluster analysis helps identify the mean differences between the clusters, which can be used to classify the data into normal and abnormal profiles. This approach is promising and can be applied to other data sets.

Moreover, the ARI assessment performed shows pair agreement between normal and aberrant clusters. Given that the

cluster partitions have been proven to be correct, we envisage a future study of device type profiling in which the output of the clustering algorithm (mean of the clusters) is entered into a `clustering-based` multivariate Gaussian outlier score (CMGOS) algorithm to classify the device type profiles and identify abnormal patterns in the data.

REFERENCES

[1] M. Kulin, C. Fortuna, E. De Poorter, D. Deschrijver, and I. Moerman, "Data-driven design of intelligent wireless networks: An overview and tutorial," *Sensors*, vol. 16, no. 6, p. 790, 2016.

[2] O. Niggemann, G. Biswas, J. S. Kinnebrew, H. Khorasgani, S. Volgmann, and A. Bunte, "Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control." in *DX@ Safeprocess*, 2015, pp. 185–192.

[3] Q. Guo, J. Luo, G. Li, X. Wang, and N. Geroliminis, "A data-driven approach for convergence prediction on road network," in *International Symposium on Web and Wireless Geographical Information Systems*. Springer, 2013, pp. 41–53.

[4] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE transactions on services computing*, vol. 9, no. 5, pp. 796–805, 2016.

[5] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data meets telcos: A proactive caching perspective," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 549–557, 2015.

[6] M. A. Muhammad, A. Ayesh, and P. B. Zadeh, "Developing an intelligent filtering technique for bring your own device network access control," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, 2017, pp. 1–8.

[7] L. A. Kurgan and P. Musilek, "A survey of knowledge discovery and data mining process models," *The Knowledge Engineering Review*, vol. 21, no. 1, pp. 1–24, 2006.

[8] W. Klösgen and J. M. Zytkow, "The knowledge discovery process," in *Handbook of data mining and knowledge discovery*. Emerald Group Publishing Limited, 2002, pp. 10–21.

[9] Ó. Marbán, G. Mariscal, and J. Segovia, "A data mining & knowledge discovery process model," in *Data mining and knowledge discovery in real life applications*. IntechOpen, 2009.

[10] I. A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *European Wireless 2014; 20th European Wireless Conference*. VDE, 2014, pp. 1–5.

[11] C. I. Eke and A. N. Anir, "Bring your own device (byod) security threats and mitigation mechanisms: Systematic mapping," in *2021 International Conference on Computer Science and Engineering (IC2SE)*, vol. 1. IEEE, 2021, pp. 1–10.

[12] P. Himthani and G. P. Dubey, "Application of machine learning techniques in intrusion detection systems: A systematic review," in *Proceedings of Third International Conference on Sustainable Computing*. Springer, 2022, pp. 97–105.

[13] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[14] P. J. Rousseeuw and M. Hubert, "Anomaly detection by robust statistics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 2, p. e1236, 2018.

[15] Q. Zhu and L. Sun, "Big data driven anomaly detection for cellular networks," *IEEE Access*, vol. 8, pp. 31 398–31 408, 2020.

[16] I. Nevat, D. M. Divakaran, S. G. Nagarajan, P. Zhang, L. Su, L. L. Ko, and V. L. Thing, "Anomaly detection and attribution in networks with temporally correlated traffic," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 131–144, 2017.

[17] J. Zhang and I. C. Paschalidis, "Statistical anomaly detection via composite hypothesis testing for markov models," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 589–602, 2017.

[18] F. Simmross-Wattenberg, J. I. Asensio-Perez, P. Casaseca-De-La-Higuera, M. Martin-Fernandez, I. A. Dimitriadis, and C. Alberola-Lopez, "Anomaly detection in network traffic based on statistical inference and\alpha-stable modeling," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 494–509, 2011.

[19] B. AsSadhan, K. Zeb, J. Al-Muhtadi, and S. Alshebeili, "Anomaly detection based on lrd behavior analysis of decomposed control and data planes network traffic using soss and farima models," *IEEE Access*, vol. 5, pp. 13 501–13 519, 2017.

[20] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 7, pp. 1460–1470, 2012.

[21] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proceedings of the joint international conference on Measurement and modeling of computer systems*, 2004, pp. 61–72.

[22] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2007, pp. 109–120.

[23] K. Xie, X. Li, X. Wang, J. Cao, G. Xie, J. Wen, D. Zhang, and Z. Qin, "On-line anomaly detection with high accuracy," *IEEE/ACM transactions on networking*, vol. 26, no. 3, pp. 1222–1235, 2018.

[24] D. Naboulsi, R. Stanica, and M. Fiore, "Classifying call profiles in large-scale mobile traffic datasets," in *IEEE INFOCOM 2014-IEEE conference on computer communications*. IEEE, 2014, pp. 1806–1814.

[25] H. Song, Z. Jiang, A. Men, and B. Yang, "A hybrid semi-supervised anomaly detection model for high-dimensional data," *Computational intelligence and neuroscience*, vol. 2017, 2017.

[26] B. Wang, S. Ying, G. Cheng, R. Wang, Z. Yang, and B. Dong, "Log-based anomaly detection with the improved k-nearest neighbor," *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 02, pp. 239–262, 2020.

[27] Y. Huang and Q. Zhang, "Identification of anomaly behavior of ships based on knn and lof combination algorithm," in *AIP Conference Proceedings*, vol. 2073, no. 1. AIP Publishing LLC, 2019, p. 020090.

[28] K. S. B. Kai, E. Chong, and V. Balachandran, "Anomaly detection on dns traffic using big data and machine learning," 2019.

[29] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," *Computer networks*, vol. 34, no. 4, pp. 579–595, 2000.

[30] shebuti Rayana, "Odds library," 2016. [Online]. Available: http://odds.cs.stonybrook.edu

[31] U. CAIDA, "Anonymized internet traces 2008 dataset," 2016.

[32] N. Eagle and A. S. Pentland, "CRAWDAD dataset mit/reality (v. 2005-07-01)," Downloaded from https://crawdad.org/mit/reality/20050701, Jul. 2005.

[33] A. S. Uluagac, "CRAWDAD dataset gatech/fingerprinting (v. 2014-06-09)," Downloaded from https://crawdad.org/gatech/fingerprinting/20140609, Jun. 2014.

[34] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "Yale: Rapid prototyping for complex data mining tasks," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 935–940. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150531

[35] M. Hofmann and R. Klinkenberg, *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

[36] A. El Attar, R. Khatoun, and M. Lemercier, "Clustering-based anomaly detection for smartphone applications," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–4.

[37] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*, 1st ed. Boston, MA, USA: Auerbach Publications, 2011.

[38] H. Liu, J. Li, Y. Wu, and Y. Fu, "Clustering with outlier removal," *arXiv preprint arXiv:1801.01899*, 2018.

[39] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[40] S. Chaimontree, K. Atkinson, and F. Coenen, "Best clustering configuration metrics: Towards multiagent based clustering," in *International Conference on Advanced Data Mining and Applications*. Springer, 2010, pp. 48–59.

[41] B. J. D. Sitompul, O. S. Sitompul, and P. Sihombing, "Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm," in *Journal of Physics: Conference Series*, vol. 1235. IOP Publishing, 2019, p. 012015.

[42] F. Tempola and A. F. Assagaf, "Clustering of potency of shrimp in indonesia with k-means algorithm and validation of davies-bouldin

index," in *International Conference on Science and Technology (ICST 2018)*. Atlantis Press, 2018.

[43] J. C. R. Thomas, M. S. Peñas, and M. Mora, "New version of davies-bouldin index for clustering validation based on cylindrical distance," in *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2013, pp. 49–53.

[44] J. M. Luna-Romera, J. García-Gutiérrez, M. Martínez-Ballesteros, and J. C. R. Santos, "An approach to validity indices for clustering techniques in big data," *Progress in Artificial Intelligence*, pp. 1–14, 2018.